

тить процесс создания новых преобразователей путем априорного отсека неоптимальных решений с позиции критериев, упомянутых выше.

Литература: 1. *I. Daubechies and W. Sweldens.* Factoring wavelet transforms into lifting steps. Technical report. Bell Laboratories. Lucent Technologies. 1996. 368p. 2. *Daubechies I. and Sweldens W.* Factoring wavelet transforms into lifting schemes. *J. Fourier Anal. Appl.* Vol. 4. 1998. P. 247–269. 3. Sweldens W. The lifting scheme: a new philosophy in biorthogonal wavelet construction. *Proc. SPIE 2569*. 1995. P. 68-79. 4. *David S. Taubman, Michael W. Marcellin.* JPEG2000: image compression fundamentals, standards and practice. Kluwer Academic Publishers. 2002. 774 p. 5. *Majid Rabbani, Rajan Joshi.* An overview of the JPEG2000 still image compression standard. *Signal Processing. Image Communication*. 2002. P. 3-48. 6. *Хаханова И.В.* Оптимизация разрядности данных для заданного качества вейвлет-преобразования Добеши // АСУ и приборы автоматики. 7. ISO/IEC 15444-1. Information Technology-JPEG2000 Image Coding System-Part 1: Core Coding System. 2000. 211p. 8. ISO/IEC 15444-2, Final Committee Draft, "Information Technology-JPEG2000 Image Coding System-Part 2: Extensions. 2000. 198p. 9. *Huang C., Tseng P., and Chen L.* Flipping structure: An efficient VLSI architecture for lifting-based discrete wavelet transform, in *Proc. IEEE Asia-Pacific Conf. Circuits and Systems*. 2002. Vol. 1. P. 383–388. 10. *Andra K., Chakrabarti C., Acharya T.* A VLSI architecture for lifting-based forward and inverse wavelet transform – *Signal Processing, IEEE Transactions on*. Vol. 50. Issue 4. 2002. P. 966-977. 11. Barua S., Carletta J.E., Kotteri K.A., Bell A.E. An efficient architecture for lifting-based two-dimensional discrete wavelet transforms. *INTEGRATION. The VLSI journal* 38 (2005). P. 341-352. 12. *Zervas N.D., Anagnostopoulos G.P., Spiliotopoulos V.,*

Andreopoulos Y., Goutis C.E. Evaluation of design alternatives for the 2-D discrete wavelet transform. *IEEE Trans. Circuits Systems Video Technol.* 11 (12). 2001. P. 1246–1262. 13. *Tinku Acharya, Ping-Sing Tsai.* JPEG2000 Standard for Image Compression. Concepts, Algorithms and VLSI Architectures. Wiley-Interscience publication. 2005. 292 p. 14. *Kotteri K. A., Barua S., Bell A. E., and Carletta J. E.* A comparison of hardware Implementations of the biorthogonal 9/7 DWT: Convolution Versus Lifting. *IEEE Transactions* 2002. P. 756 - 767. 15. www.mathworks.com 16. *Хаханов В.И., Хаханова И.В.* VHDL + Verilog = Синтез за минуты. Харьков: СМИТ. 2007. 264 с. 17. *Семенец В.В., Хаханова И.В., Хаханов В.И.* Проектирование цифровых систем с использованием языка VHDL. Харьков: ХНУРЭ. 2003. 492 с. 18. *Charles H. Roth, Jr.* Digital Systems Design Using VHDL. Boston. PWS Publishing Company. 1998. 470 p. 19. *Ashenden, Peter J.* The designer's guide to VHDL. San Francisco. Calif. California. Morgan Kaufmann Publishers, Inc. 1996. 688 p. 20. Xilinx.com.

Поступила в редколлегию 30.11.2007

Рецензент: д-р техн. наук, проф. Кривуля Г.Ф.

Побеженко Ирина Александровна, ассистент Государственной академии культуры. Научные интересы: проектирование цифровых систем на кристаллах. Увлечения: английский язык, музыка. Адрес: Украина, 61166, Харьков, ул. Плехановская, 73-85 тел. 7374287. E-mail: irina_pob@ukr.net

Хаханова Ирина Витальевна, докторантка кафедры АПВТ ХНУРЭ. Научные интересы: проектирование цифровых систем на кристаллах. Увлечения: английский язык, музыка. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 70-21-326. E-mail: hahanova@mail.ru

УДК004.415.2

ИССЛЕДОВАНИЕ АРХИТЕКТУРЫ СЕРВИСНО-ОРИЕНТИРОВАННЫХ СИСТЕМ

ГОРБАЧЕВ В.А., ЕФАНОВ А.Ю.

Разрабатывается принцип оптимального функционального взаимодействия между компонентами сервисно-ориентированной системы, а также обеспечения эффективной и надежной работы всей компьютерной системы, построенной на базе технологий виртуализации и кластеризации. Оценивается эффективность использования технологий виртуализации, ее преимущества и недостатки.

Актуальность, новизна и цель работы

За последнее время развития компьютерных систем было разработано и испытано несколько перспективных технологий, которые могут быть использованы как для улучшения характеристик существующих систем, так и для устранения некоторых их недостатков. Среди недостатков существующих компьютерных систем можно отметить:

– Невысокие показатели стабильности работы, длительное время, требуемое для восстановления после сбоя, а также нерациональное и неэффективное использование оборудования.

– Большие сложности в интеграции нескольких компьютерных систем между собой.

– Ограниченные возможности доступа к ресурсам системы с мобильных устройств.

– Низкая гибкость существующих систем.

Приведенные выше недостатки могут быть устранены при проектировании компьютерной системы на базе следующих технологий.

Виртуализация. В данном контексте виртуализация – это возможность запуска нескольких операционных систем (виртуальных машин) на одном физическом компьютере. Применение технологий виртуализации предоставляет возможность более эффективно использовать существующее оборудование, а также повысить общую гибкость компьютерной системы [1].

Кластеризация. Компьютерный кластер – это группа компьютеров, объединенных между собой и работающих вместе так, что представляют собой единый и унифицированный компьютерный ресурс. Технологии кластеризации позволяют повысить характеристики стабильности работы, а также снизить время, необходимое для восстановления после сбоя.

Сервисно-ориентированная архитектура (СОА). СОА – метод разработки программного обеспечения, базирующийся на использовании сервисов (служб). Приложения, разработанные с применением данной

архитектуры, представляют собой универсальные компоненты, обладающие высокой гибкостью работы и широкими возможностями для взаимодействия через сеть. Такие приложения позволяют повысить гибкость компьютерной системы, а также предоставляют большие возможности по интеграции как программ, так и компьютерных систем между собой.

Повышение характеристик существующих компьютерных систем, расширение их функциональности, поиск новых методов их применения, а также устранение определенных их недостатков позволяет сделать вывод об актуальности исследований в этой области.

В работе объектом исследования является архитектура сервисно-ориентированных компьютерных систем.

Цель работы – создание эффективной архитектуры сервисно-ориентированной системы, функции которой могут быть доступны с мобильных устройств, с настольных компьютеров, а также из других компьютерных систем.

Задачами исследования являются:

– Создание высокопроизводительной, надежной, гибкой аппаратно-программной архитектуры компьютерной системы.

– Экспериментальная оценка производительности исследуемой архитектуры.

Выбор аппаратно-программной архитектуры компьютерной системы

Использование рассмотренных технологий в комплексе позволяет проектировать системы, лишенные указанных выше недостатков, а также имеющие улучшенные характеристики и расширенную функциональность, что необходимо при работе систем в промышленных условиях.

Общая модель проектируемого комплекса базируется на компьютерном кластере, объединяющем несколько компьютерных серверов и одно хранилище данных, которое также может быть выполнено в виде сервера. Компьютерный комплекс работает под управлением операционной системы Microsoft Windows 2003 Enterprise Edition и использует службу Microsoft Clustering Services для своего функционирования.

На каждом из узлов кластера установлена система виртуализации Microsoft Virtual Server 2005 R2, которая содержит виртуальные машины и занимается их управлением. Каждая виртуальная машина подключена к открытой сети и предоставляет доступ к определенным службам (база данных, сервер приложений, веб-сервер и т.д.). Основным критерием выбора данной системы виртуализации является наличие базовой поддержки работы в кластере.

Программное обеспечение, занимающееся управлением виртуальными машинами, а также службами, которые на них выполняются, может располагаться

как на физической, так и на виртуальной машине. Данное приложение также занимается управлением задач, которые выполняет компьютерный кластер, и определением возможностей доступа к исполняемым на комплексе ресурсам.

Кластерная архитектура, лежащая в основе данной компьютерной системы, была выбрана, в первую очередь, для обеспечения отказоустойчивости, а также для повышения производительности системы. Благодаря ее использованию система может балансировать нагрузку между компьютерами, входящими в кластер, а также, в случае возникновения отказа аппаратного обеспечения либо критической программной ошибки, продолжить выполнение задач на работающем узле. Задача в данной компьютерной системе представляет собой виртуальную машину, что повышает гибкость системы и позволяет эффективно использовать аппаратное обеспечение. Благодаря использованию данных технологий компьютерная система может перемещать виртуальную машину с одного узла на другой для балансирования нагрузки, а также в случае отказа какого-либо узла. Для повышения производительности возможно выключение какой-либо виртуальной машины, когда она не используется.

Весь комплекс может быть подключен к корпоративной сети, либо к сети Интернет. Доступ к проектной системе может осуществляться как с настольных компьютеров, так и с мобильных устройств.

Общая модель комплекса изображена на рис. 1. На ней представлены два сервера, на которых запущены виртуальные машины, а также общее хранилище данных, выполненное в виде сервера. Компьютеры подключены к закрытой сети (для поддержки работоспособности кластера), а также к открытой (для взаимодействия с различными устройствами).

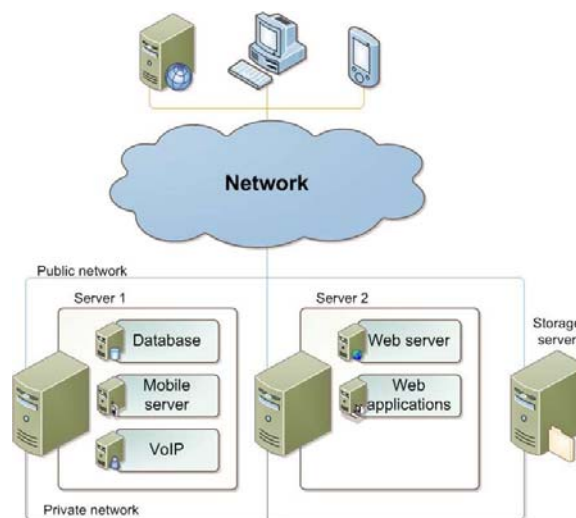


Рис. 1. Общая модель системы

Планирование и проведение машинного эксперимента

Одной из задач создания эффективной аппаратно-программной архитектуры системы является повышение эффективности использования аппаратного обеспечения, увеличение гибкости компьютерной системы. Для определения возможности использования технологий виртуализации для решения поставленных задач был проведен эксперимент.

Целью данного эксперимента являлась оценка эффективности использования технологий виртуализации при проектировании архитектуры сервисно-ориентированной системы на ее базе. Для выполнения цели эксперимента был проведен анализ производительности приложения, работающего как на физическом компьютере, так и на виртуальном.

Объектом исследования был программный комплекс, состоящий из веб-приложения, представляющего собой веб-сайт, и сервисно-ориентированного компонента, являющегося веб-сервисом и предоставляющего данные для него.

Для достижения целей исследования было выбрано нагрузочное тестирование, которое заключалось в создании возрастающей нагрузки на приложение и, следовательно, на веб-сервис. Тестирование состояло из симулирования посещения веб-приложения пользователями по единому во всех случаях маршруту. На протяжении теста количество пользователей возрастало от 0 до 2000 с возрастанием на 5 каждые 10 секунд. Время выполнения каждого теста составило 1 час 6 минут. Во время тестирования замерялись характеристики, которые могут свидетельствовать о производительности работы сервера приложений, а также операционной системы в целом. Тестирование производилось на двух компьютерах с размещенным тестовым комплексом в реальной операционной системе и в виртуальной системе. Технические характеристики компьютеров представлены в табл. 1.

Меньший объем памяти на виртуальной машине вызван ограничением, накладываемым системами виртуализации, так как оперативная память также необходима операционной системе для поддержки функционирования системы виртуализации. Следовательно, результаты работы приложений в виртуальной среде будут заведомо хуже, чем при запуске на физических компьютерах. Также из исходных условий можно определить более высокую нагрузку на компьютерную сеть (закрытую), связанную с необходимостью ее частичной эмуляции для предоставления доступа к ней из виртуальных операционных систем. Следовательно, сложно сравнивать общую производительность между физическими и виртуальными машинами.

Основными отличиями приведенных систем являются поддержка аппаратной виртуализации у AMD64, а также наличие двух ядер у процессора CORE2.

Тестирование комплекса осуществлялось с помощью программы для тестирования веб-приложений

Технические характеристики компьютеров

Характеристика	CORE2	AMD64	VIRTUAL
Тип компьютера	Физ.	Физ.	Вирт.
Процессор	Intel Core 2 Duo T5500	AMD Athlon 64 3000+	Тот же что и на физ. машине
Поддержка аппаратной вирт.	Нет	Есть	---
Аппаратная платформа	32 бита	64 бита	---
Количество реальных или виртуальных ядер	2 ядра	1 ядро	1 ядро
Объем оперативной памяти	1024 Mb	1024 Mb	600 Mb
ОС	Windows 2003 Enterprise Edition	Windows 2003 Enterprise Edition	Windows 2003 Enterprise Edition
Сервер приложений	IIS 6.0	IIS 6.0	IIS 6.0
Пропускная способность сетевого канала	100 Мбит	100 Мбит	100 Мбит

WAPT 4.0 и оценивалось по трем основным параметрам.

Среднее количество страниц в секунду (Average Pages per Second) – усредненная характеристика, показывающая количество уникальных страниц, генерируемых сервером по запросу пользователя в секунду на протяжении всего теста.

Среднее время отклика при ~2000 пользователей (Average response time) – время между запросом пользователя и ответом сервера. Приведенное в табл. 1 время отклика является усредненной характеристикой при максимальной нагрузке, т.е. около 2000 запросов от пользователей одновременно.

Скорость получения данных при ~2000 пользователей – скорость, с которой сервер отправляет данные клиентам. Приведенная характеристика также является усредненным параметром при максимальной нагрузке.

График общей производительности представляет собой объединение всех перечисленных параметров, измеряемых при тестировании. График позволяет оценить общую производительность компьютерной системы, а также максимальную нагрузку, которую он способен выдержать. На рис.2, 3, 4, 5 изображено 3 параметра:

Количество активных пользователей. Данный параметр указывает на нагрузку сервера, выражающуюся в количестве пользователей, которых сервер должен обслуживать. Параметр позволяет оценить поведение других характеристик под различной нагрузкой. На графике параметр представлен черной линией с квадратами и значениями на вертикальной оси справа.

Количество запросов в секунду. Данный параметр позволяет оценить, какую нагрузку способен выдержать сервер на данном программном и аппаратном обеспечении. Параметр зависит от производительности сервера, следовательно, позволяет оценить возможности процессора. Параметр представлен серой линией с кружками и шкалой значений слева.

Количество страниц в секунду. Данный параметр также позволяет оценить предельную нагрузку сервера, но в отличие от предыдущего параметра он в большей степени зависит от оперативной памяти. На графике он представлен зеленой линией с ромбами и значениями слева.

Время тестирования. Горизонтальная ось на графике – это ось времени, которая показывает количество времени, прошедшее с начала тестирования.

Анализ результатов тестирования на физических компьютерах

В первой части производится сравнение результатов тестирования программного комплекса, расположенного на различных физических машинах без использования виртуализации.

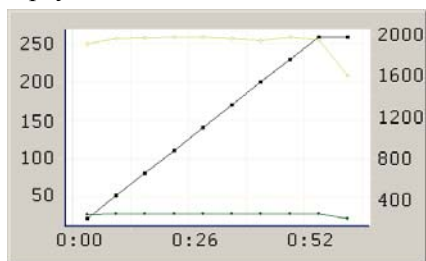


Рис. 2. Общая производительность CORE2

На рис. 2 видно, что сервер на базе процессора Core 2 стабильно выдерживает нагрузку с возрастающим количеством пользователей, однако при возрастании количества активных пользователей до 2000 системных ресурсов становится недостаточно. Об этом свидетельствует уменьшение количества обрабатываемых запросов в секунду с 250 до 205. Стабильный показатель среднего количества страниц в секунду обуславливается тем, что они кэшируются (сохраняются в памяти сервера IIS) и выдаются всем пользователям одинаковые, так что времени на генерирование страниц не требуется, если их содержимое не меняется.

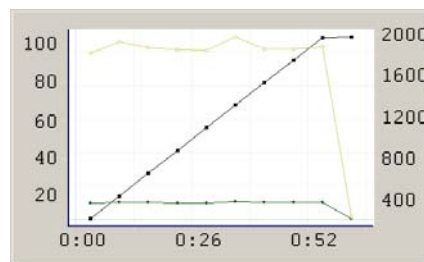


Рис. 3. Общая производительность AMD64

На рис. 3 прослеживается аналогичная ситуация с процессором AMD64, как и с процессором CORE2, однако следует отметить, что количество обрабатываемых запросов в секунду у AMD64 равно 100, с незначительными увеличениями на протяжении теста. Также следует отметить более резкое падение количества обрабатываемых запросов при увеличении количества активных пользователей до 2000. Количество генерируемых страниц, как и в первом случае, достаточно стабильно, что также обусловлено возможностями кэширования у сервера приложений IIS.

Компьютер с процессором Intel Core 2 Duo показывает значительное превосходство над рабочей станцией, основанной на процессоре AMD Athlon 64. Среднее обрабатываемое количество страниц в секунду у CORE2 практически в 3 раза больше, чем у AMD64, а также практически в 2 раза более низкое время отклика системы. Скорость передачи и получения данных у CORE2 более чем в 2 раза выше по сравнению с AMD64 (табл.2).

Значительно более высокую производительность у сервера на базе процессора Intel Core 2 Duo можно

Таблица 2

Числовые характеристики теста на физических машинах

Характеристика	CORE2	AMD64
Среднее количество страниц в секунду	28,4	10,1
Время отклика при ~2000 пользователей	38,2	50,3
Скорость получения данных при ~2000 пользователей	29,2	13,8

объяснить наличием у него двух ядер, а также оптимизацией сервера приложений IIS 6 под многопоточную среду, которая эффективно использует наличие двух процессоров или двух ядер в системе.

Анализ результатов тестирования на виртуальных компьютерах

Во второй части тестирования рассмотрена производительность тестового комплекса в виртуальной среде, которая также обеспечивается на различных аппаратных платформах.

Числовые характеристики теста на виртуальных машинах

Характеристика	Virtual CORE2	Virtual AMD64
Среднее количество страниц в секунду	5,7	9,7
Время отклика при ~2000 пользователей	40,1	64,7
Скорость получения данных при ~2000 пользователей	4,3	11,6

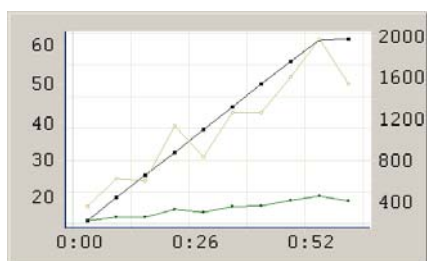


Рис. 4. Общая производительность виртуальной машины CORE2

На рис. 4 видно, что виртуальный сервер на базе процессора CORE2 работает достаточно нестабильно, так как количество обрабатываемых запросов в секунду очень динамично, также видно заметное уменьшение данного параметра с 250 до 60. Такое уменьшение обуславливается использованием только одного ядра данного процессора, что является ограничением системы виртуализации. Количество генерируемых страниц в данном случае не статическая величина, что обуславливается различным количеством запросов в секунду, так как не возникает необходимости в хранении всего сайта в памяти.

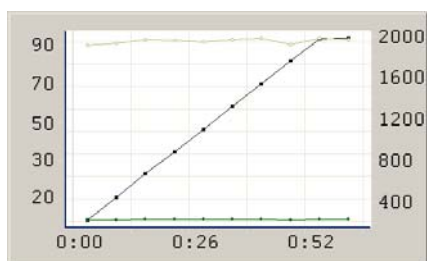


Рис. 5. Общая производительность виртуальной машины AMD64

Результаты теста виртуальной машины на базе AMD64 (рис.5) очень похожи на результаты тестирования на физической машине, из отличий следует отметить только небольшое снижение количества обрабатываемых запросов в секунду. Столь большое отличие от результатов теста процессора CORE2 объясняется аппаратной поддержкой виртуализации. Отсутствие серьезного падения производительности также обуславливается наличием только одного ядра у процессора AMD64. Следует отметить стабильность работы данного процессора в виртуальной среде.

В результатах теста, указанных в табл. 3, среднее количество страниц в секунду у процессора Intel Core 2 Duo уменьшается почти в 7 раз, тогда как у процессора AMD оно падает на 8%. Время отклика у обоих процессоров возрастает на 15-20%. Скорость получения данных при максимальной нагрузке у процессора Intel также ощутимо падает, как и среднее количество страниц в секунду, т.е. приблизительно в 7 раз, тогда как у виртуальной машины на процессоре AMD этот показатель уменьшается на 9%.

Отличные результаты теста виртуальной машины на процессоре AMD Athlon 64 объясняются наличием аппаратной поддержки виртуализации, тогда как процессор Intel Core 2 Duo T5500 не поддерживает виртуализацию на аппаратном уровне. На результаты работы виртуальной машины CORE2 также оказывает отрицательное влияние факт использования при виртуализации только одного ядра процессора, так как система виртуализации Virtual Server 2005 не может использовать больше одного процессора для эмуляции одной виртуальной машины, а для операционной системы наличие двух ядер эквивалентно наличию двух процессоров.

Следует еще раз подчеркнуть, что объем виртуальной памяти, доступный виртуальным машинам, был 600 Мб, что на 400 Мб меньше, чем тот, что был доступен при прохождении теста физическим машинам.

Из результатов эксперимента можно сделать вывод о том, что использование систем виртуализации оправдано в компьютерных системах. Падение характеристик на 15% незначительно по сравнению с преимуществами, которые несет данная технология: более рациональное и эффективное использование оборудования, быстрое восстановление после сбоя. Также следует отметить, что перенос в виртуальную среду имеет смысл для систем, работающих не на полную нагрузку. Как показывают исследования Gartner [2], в инфраструктуре средней организации серверы находятся под нагрузкой в 10 – 30% от их максимальной производительности. Перенос таких систем в виртуальную среду практически не скажется на их производительности и не будет заметен для клиентов сервера, но позволит снизить количество необходимого аппаратного обеспечения.

Выводы

В исследовании наглядно показаны преимущества использования процессоров с аппаратной поддержкой виртуализации, с применением которых производительность тестового программного комплекса падает всего на 10-15%, что сравнимо со снижением производительности физических компьютеров при уменьшении количества установленной у них оперативной памяти.

Из исследования можно сделать вывод, что для работы больших программных комплексов можно реко-

мендовать использование процессоров с аппаратной поддержкой виртуализации, т.е. поддерживающих технологию AMDV у процессоров AMD, либо технологию IntelVT у процессоров фирмы Intel. Дополнительное увеличение производительности можно получить при использовании многоядерных процессоров, так как каждое из ядер может быть задействовано определенной машиной.

Научная новизна и практическая ценность. Технологии виртуализации позволяют более эффективно и рационально использовать аппаратное обеспечение, а также значительно повысить гибкость компьютерной системы. Из полученных результатов исследования, а также проведенного эксперимента можно сделать вывод о практической ценности использования технологий виртуализации в компьютерных системах для повышения гибкости и надежности компьютерной системы, а также для более рационального и эффективного применения аппаратного обеспечения.

Литература: 1. *David Marshall, Wade A. Reynolds.* Advanced Server Virtualization: VMware and Microsoft Platforms in the Virtual Data Center. AUERBACH, 2006 2. Gartner inc. <http://www.gartner.com>, 2007-2008.

Поступила в редколлегию 22.12.2007

Рецензент: д-р. техн. наук, проф. Руденко О.Г.



Горбачев Валерий Александрович, канд. техн. наук, профессор каф. ЭВМ ХНУРЭ. Научные интересы: моделирование систем. Хобби: музыка, волейбол, автомобили. Адрес: Украина, 61136, Харьков, ул. Героев Труда, 33, кв. 300.



Ефанов Александр Юрьевич, студент ХНУРЭ. Научные интересы: сервис-ориентированные системы, компьютерные сети. Адрес: Украина, 61071, Харьков, ул. Кибальчича, 33, кв. 8, тел. (057)376-16-36, e-mail: deslocator@gmail.com.

УДК519.713:681.326

СИСТЕМНАЯ МОДЕЛЬ АППАРАТНОЙ РЕАЛИЗАЦИИ MQ-КОДЕРА

ХАХАНОВА И.В.

Предлагается системная модель аппаратной реализации MQ-кодера, выполненная с помощью редактора Simulink [1-7]. Управление работой алгоритма реализуется с помощью блока, представляющего сеть из семи параллельных автоматов. Архитектура управляющего модуля, состоящая из нескольких автоматов с небольшим количеством состояний, позволяет повысить частоту синхронизации устройства, а значит и скорость кодирования данных.

1. Введение

MQ-кодер реализует алгоритм адаптивного арифметического сжатия [8 - 15] путем кодирования наиболее и наименее значимых данных на основе пары [D – значение, CX – контекст] сгенерированных алгоритмом EBCOT в результате сканирования битовых плоскостей.

Основой устройства являются два регистра, определяющие параметры интервала кодирования. Регистр А содержит значение текущего интервала, а регистр С хранит частичное слово для процесса кодирования. На начальном этапе регистр А инициализирован значением 0x00008000, которое обозначает начальную внутреннюю вероятность, а в регистр С записывается код 0x00000000, идентифицирующий отсутствие сгенерированных слов сжатой последовательности. На рис. 1 представлен формат регистров А и С. Биты с меткой "а" соответствуют дробной части регистра А (текущее значение интервала), биты с меткой "х" – дробной части регистра С. Биты, отмеченные символом "s" – это разделители, блокирующие возмож-

ность возникновения переноса в старшие разряды. Биты "b" обозначают позицию готового сжатого кода, который переносится из регистра С в память. Бит "с" – бит переноса.

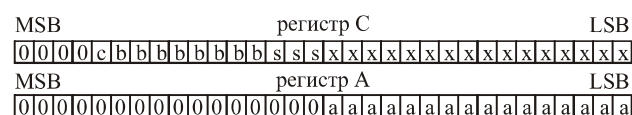


Рис. 1. Структура регистров MQ-кодера

2. Архитектура устройства

Алгоритм сжатия данных в стандарте [11, 12, 14, 15] представлен в виде главной процедуры Encoder, вызывающей подчиненные ей функции. Иерархия процедур MQ-кодера представлена на рис. 2.

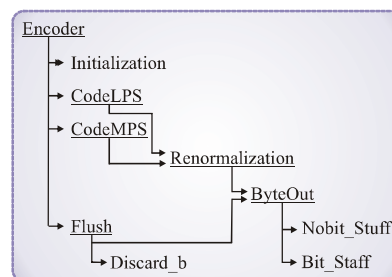


Рис. 2. Иерархия процедур алгоритма MQ-кодера

Предлагаемая модель аппаратной реализации MQ-кодера (рис. 3) включает иерархический управляющий автомат FSM, основанный на 7 компонентных автоматах, и операционный автомат ALU, состоящий из двух блоков: АЛУ регистров (ALU Register) и АЛУ условий (ALU Condition). Simulink-модель устройства представлена на рис. 4. На основе программного алгоритма MQ-кодера составлены таблицы основных операций устройства (табл. 1) и условий (табл. 2). В зависимости от регистра, над которым выполняется